

Manual for data-analysis with Delft-AUKE

The process of extracting the right parameters out of some raw data

Arjen Koopman

April, 2006

Contents

1. Introduction	3
2. Structure of Delft-AUKE.....	4
3. Installation of Delft-AUKE	5
4. How to use Delft-AUKE?	6
4.1. Converting the raw data with excel	6
4.2. Batch files	6
4.3. Conasc	7
4.4. Waves	9
4.5. Spectrum	11
4.6. Extracting results from the parameter files	12
5. Checklist.....	14
A. Macro change files	15
B. Macro extract data.....	18

1 Introduction

This document is a manual that describes the use of Delft-AUKE for the analysis process of measurements from pressure sensors, optical backscatter sensors (OBS) for sediment concentration and electromagnetic water flow sensors (EMF). In this manual we will go step by step through the process of extracting the right parameters out of some raw data. This will not only be done with Delft-AUKE but also Microsoft Excel will be used. With this manual several files will be delivered. The following files are needed:

- results.xls (excel file for changing files and extracting parameters)
- golven.bat (batch file for a quicker process)
- conasc.pcf (command file for CONASC)
- waves.pcf (command file for WAVES)
- spectrum.pcf (command file for SPECTRUM)
- and of course Delft-AUKE

This manual contains three chapters. In the first chapter there will be told something about how Delft-AUKE is structured. The second chapter will tell you how you can install Delft-AUKE on your computer. The last chapter will handle the process of extracting parameters out of some raw data. The subjects that will be explained are:

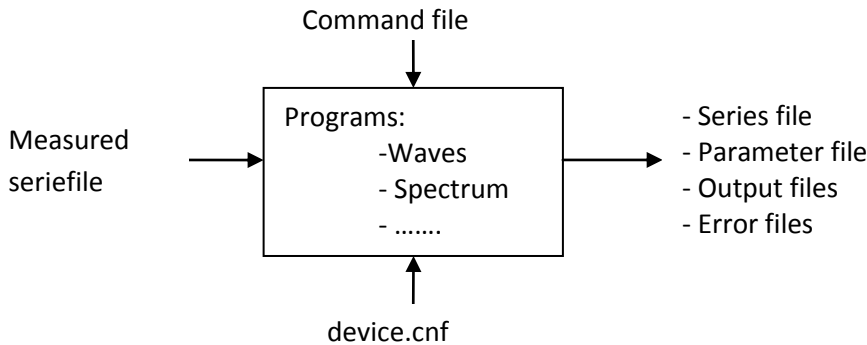
- Converting the raw data to files that Delft-AUKE can use as input files.
- Making batch files to do many data files in one time.
- The use of the Delft-AUKE programs CONASC, WAVES and SPECTRUM.
- Extracting the needed parameters from the output files from Delft-AUKE.

2 Structure of Delft-AUKE

DELFT-AUKE is a set of programs for processing of signals from (laboratory) instruments.

The program takes measured data files as input, and following a detailed set of instructions in a so-called .pcf command file, generates output in form of new series files, parameter tables, plots and other output files.

All processing programs are structured in the same way, visualized below:



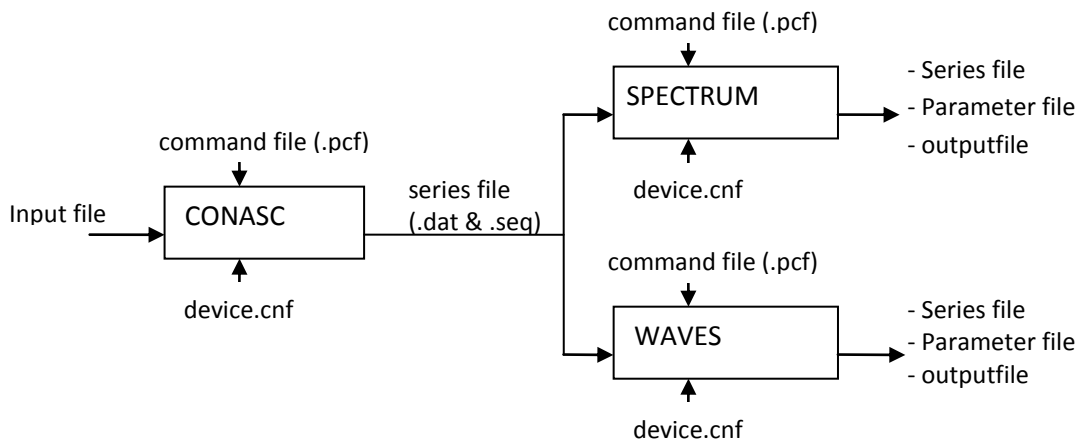
The measured series file is the measured data in Delft-AUKE format. This means that the raw data files from the field measurements are not suitable as input for Delft-AUKE. The raw data has to be converted with the Delft-AUKE program CONASC. As a result of this conversion a data file with extension .dat and an information file with extension .seq will be created.

The command file instructs the program what to do. The device.cnf file is a configuration file which specifies environment information which should not be changed.

The output can be a new series file which can be used by other programs.

The program also generates parameter files (with extension .PAR), the output file, which is a text file with all the instructions that have been used, and – if an error occurred – the error file.

The process structure of Delft-AUKE for this manual will be as follows:



3 Installation of Delft-AUKE

Delft-AUKE is not more than a folder with the executables and system files in it. If you want to run a program somewhere on your computer the system needs to know in what directory Delft-AUKE executables and system files are stored. For that reason an environment variable AUKEPC must be set that points towards that directory. You can set this variable as follows:

Go to the START menu on the taskbar.

Go to control panel → system → advanced-tab → Environment Variables-button.

Here you will see two windows. The first is called user variables and the second is called system variables.

In the top window (user variables) click on 'new'. Here you have to fill in a name, for example Aukepc, and a variable value. This value is the directory where the executables and system files are stored. (For example: c:\aukeprocessing)

Also, add this directory to the PATH environment variable in the second window. This can be done by editing the PATH variable and paste the directory name behind the others. Make sure you put a ';' in front of it to separate it from the other directory names. And don't delete any of these other names as well.

When these changes have been done you must restart your PC to make them work.

The programs have been installed in a home directory. The file c:\autoexec.bat has been changed so that the programs in this directory will be used even if the set of instructions is in a different working directory.

4 How to use Delft-AUKE?

4.1 Converting the raw data with excel

Before Delft-AUKE can be used, the data files have to be changed in such a way that the Delft-AUKE program CONASC can read them. The data files you start with are in a comma separated value format (.csv). The changing of the files is something you can do with excel, with the use of a Visual basic macro. To use a macro you have to make sure the security level is set correctly. To check this open excel and go to Tools → Options → Security → Macro security. Here you have to set the security level at medium.

The excel document that can be used is delivered together with this manual and is called '*results.xls*'. For changing the data files the sheet '*change files*' has to be opened. You will see some yellow cells where you have to fill in some standard values, before you can run the macro. The comments in front of the yellow cells will tell you what to do. After filling in this sheet you can run the macro by clicking the button. Behind the button there is a macro (Appendix A) inside this excel document that will change the following things in the data:

- Adding a time step instead of the first column with the label number
- Delete the header and ending rows.
- Ignore empty files or files without enough information.
- Make a batch file for running AUKE.

The only thing this macro won't do is checking for corrupt files. It is possible that you have some files where the recording process went wrong. Mostly you can filter these files out by checking the file size. So before you can run the macro you have to check for corrupt files. You have to do this before running the macro, because else the batch file won't be correct.

4.2 Batch files

Delft-AUKE can only handle one data file at a time. But because almost every time there will be more than one data file, batch files will be used. The first batch file is the one that was created together with the input data files. This batch file will make it possible to run all the data files that were created after each other. This batch file will have the name *startxxx.bat* and it will look like this:

```
call golven.bat xxxx001
call golven.bat xxxx002
call golven.bat xxxx006
call golven.bat xxxx007
call .....
```

Call golven.bat means that the file *golven.bat* will be opened. *Golven.bat* is also a batch file. The last command in every line is the file name of the data file that should be opened, and it is stored as '%1'. For every data file there will be a new line.

The file *golven.bat*, that is reverved to is delivered with this manual. This looks as follows:

```
copy conasc.pcf input.pcf
mod -s ABCD %1 input.pcf
```

```
conasc in=input.pcf ou=%1
```

```
copy waves.pcf input.pcf  
mod -s ABCD %1 input.pcf  
waves in=input.pcf ou=%1
```

```
copy spectrum.pcf input.pcf  
mod -s ABCD %1 input.pcf  
spectrum in=input.pcf ou=%1
```

The first line will copy the command file *conasc.pcf* to *input.pcf*. This is done to change the command file, but leave a general command file unchanged. The second line will change every ABCD in the command file and replace it with %1, which is the name of the data file. Finally the third line will activate the Delft-AUKE program CONASC with *input.pcf* as input and an output file with the name of the data file. Then the same is done for the Delft-AUKE programs WAVES and SPECTRUM. When this is finished *golven.bat* will go on with the next data file.

4.3 Conasc

Now you have created the input file, working with Delft-AUKE can be started. Before you can run the WAVES and SPECTRUM program you have to convert the data files to Delft-AUKE series files. Only then WAVES and SPECTRUM will be able to read the data. This conversion can be done with the Delft-AUKE program CONASC. In order to run this program a command file (.pcf) has to be used. With this manual the command file *conasc.pcf* is delivered. The command file looks like this.

```
** conasc.pcf **
```

```
datafile,ABCD.txt  
data,conABCD,delete  
FILE-TYPE,FREE  
#heading-lines,0  
#end-lines,0  
get-columns, 1:time, 2:pres1, 3:pres2, 4:pres3, 5:pres4, 6:obs1, 7:ems1x, 8:ems1y, 9:obs2,  
10:ems2x, 11:ems2y  
separator,""  
equidistant-base,time
```

There may be some things that have to be changed in this file. If you didn't take the same number of rows for every file, you have to change the number of ending rows in 'conasc.pcf', if there are any ending rows. You also have to look at the command line get-columns. In this line you can give names to the different columns. If something isn't right you should change it.

Explanation of the command lines:

datafile,ABCD.txt

With this command the name of the input data file is given. This is the file that was created with the excel macro. Here it is called ABCD.txt, because it is a .txt file. ABCD is the name you can change with a batch file. (see chapter 4.2)

data,conABCD,delete

This command is where the name of the output series files is set. No extensions must be given; the program itself adds the extensions for the data file (.dat) and the information file (.seq). Every output file will be named conABCD. The delete command at the end will delete existing files with the same name.

FILE-TYPE,FREE

With this command the file type of the input file is selected. An input file can be in Delft-AUKE format. This is the case when the data files have the same layout as a file generated with the Delft-AUKE program ASCII. But because the .txt file was created with excel this is not the case. These files are in the so called FREE format.

#heading-lines,0

When the file type is free the program needs to know how many lines precede the data lines. These lines are skipped and data reading starts after the given number of lines. In this case the number of heading lines will be zero, because the excel macro already deleted the heading lines.

#end-lines,0

When the file type is free it is allowed that some lines at the end of the data lines are not used. These lines can be either data lines or comment lines. When this option is used the program must read the complete file two times, so this option will take more time. Just like the heading lines, the ending lines are already deleted by the excel macro. But if you want to delete some data lines as well this figure has to be changed.

get-columns,1:time,2:pres1,3:pres2,4:pres3,5:pres4,6:obs1,7:ems1x,8:ems1y,9:obs2,10:ems2x,11:ems2y

This command is used to indicate what series (=columns) must be written to the new file. If you want to store columns 1, 2 and 5 then the command will look like: get-columns,1,2,5. In the series file the names of the series are set to COL followed by the ordering number. So in this example the names will be, COL01, COL02 and COL05. These series can also be named. This can be done by following the column number with a name you want to give that column. This is done in this command file. It is possible that these names don't match the data that you have. In that case you will have to change this command line.

separator,""

This is to indicate how the different columns are separated in the input file. The columns in the text file that was created are separated by a comma.

equidistant-base,time

Here the column is selected on which the other data is based. This is of course the time column.

4.4 Waves

With WAVES the development of series values between two positive zero crossings is determined. Any signal that acts such as a wave signal can be used. The command file (*waves.pcf*) looks like this:

```
** waves.pcf **  
  
datafile,conABCD  
data,wABCD,delete  
scale,yes  
zerolevel,no  
series,all  
store,waves  
use  
poly2,calculate  
end:use  
par-file,wABCD  
parameter  
all  
end:parameter
```

In this command file you don't have to change anything. You can use it the state it is delivered.

Explanation of the command lines:

datafile,conABCD

With this command the name of the input series file is given. These are the .dat and .seq files created by CONASC. ABCD is the name you can change with a batch file. (see chapter 4.2)

data,wABCD,delete

This command is where the name of the output series files are set. No extensions must be given; the program itself adds the extensions to the data file (.dat) and the information file (.seq). Every output file will be named wABCD. The delete command at the end will delete existing files with the same name.

scale,yes

With this command the user lets the program know whether or not a scale factor must be used. This factor is stored in the output .seq file from CONASC. If you use the scale command but the directive is not found in the information file then no scaling will be performed. In this case this command has no effect, because the scale factor in the .seq file is equal to 1.

zerolevel,no

When an instrument is used in a scale model the zero value of the signal and the coordinate of the position where the signal is measured may differ. To bring these quantities into conformity one with another a zero level value can be set. This value is stored in the series information file (SEQ). Another possibility is that zero is far away from any signal value and that a special zero level must be used in processing a series. With the command zerolevel a program can be informed whether this value must be used or not. If it is used the zero level value will be subtracted from each stored series value: the signal values are related to the zero level value. This new value is then used for the computation. In this case, to get zero-crossings, the use command is used so the zerolevel is not necessary.

series,all

Here the series that have to be used are selected. In this case all series are selected. The series are the different columns that where each given a name in the CONASC program.

store,waves

A number of resulting quantities can be stored in a new file. The number of series is the same as the number of store commands. A quantity is a series with as much values as the number of waves in the original series. These quantities can be:

Waves - The wave heights found.

t-waves - The starting location of the wave.

Period - The period of the wave.

Crest - The highest value in the crest

t-crest - The location of the crest

Trough - The lowest value in the wave.

t-trough - The location of the trough.

But here only waves is stored. The stored series file can be used for plotting with AUKEPLOT. More about AUKEPLOT can be red in the official manual.

use**poly2,calculate****end:use**

The parameter values within the use block give some statistical information which can be used by the processing programs. One of these parameter values is the “poly” command. Data values can be related to a level that is defined by a polynomial. A polynomial has the general formula:

$$y(j) = \sum_{k=0}^3 c_k x^k(j)$$

With j the ordering number of the data values in the series. The computed value for a data value according to the used polynomial is subtracted from the measured data value. This way a possible trend in the series is removed. The number that is attached to poly gives the order of the polynomial. So the poly0 polynomial equals the mean of the selected series part. poly1 will subtract a linear trend and poly2 will subtract a parabolic trend. The poly2 is used here, because the tide signal has to be subtracted before the waves can be calculated. This is because the waves are calculated with the use of upward and downward crossings of the zero line. In figure 1 the data from one pressure sensor during one measurement is shown. You can see that because of the influence of the tide there aren't any zero crossings. That's why a second order polynomial (the black line) is calculated and subtracted. The black line will become the new zero line.

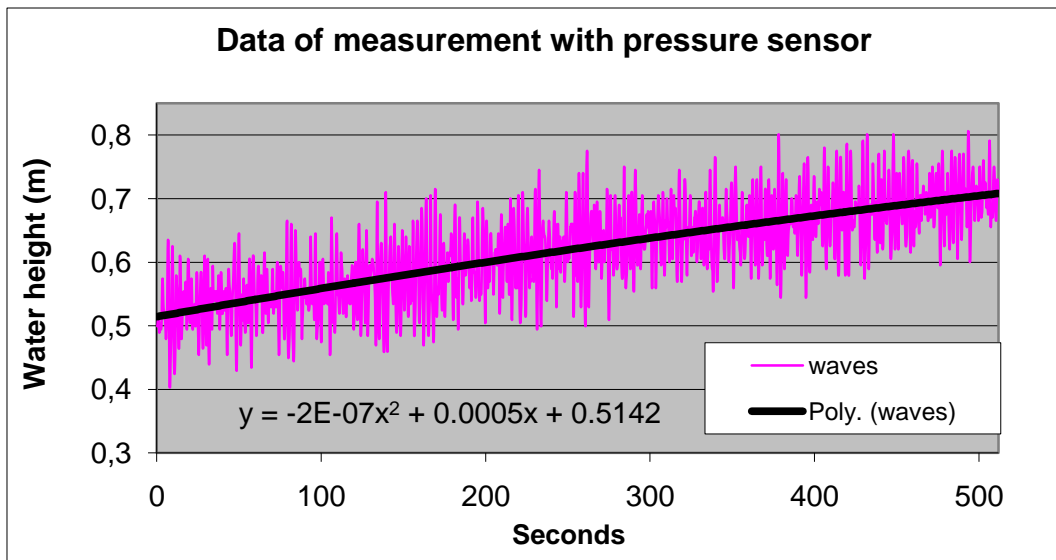


fig 1.: Data from a pressure sensor of a random measurement of the Zuidgors. The formula that is shown belongs to the polynomial.

par-file,wABCD

Here the name of the .par files is set. Every parameter file will be named wABCD.

parameter

all

end:parameter

This block is used to select the parameters that should be stored in the parameter file. Here the command all is used, which means that all the parameters available will be stored.

4.5 Spectrum

With the program SPECTRUM the spectral densities of series are computed. A number of series may be selected, up to the number of series available in the data file. Together with the densities a set of parameters is computed and sent to the chosen output device. The spectral density series can be sent to a new series file after which this file can be applied for plotting or for further post processing.

It is possible that SPECTRUM will give an error. This mostly is because the data in a specific column is almost the same in every row. What you can do is change a couple of values or don't select that column with CONASC, because if the values are all the same you probably don't need that data anyway.

The command file doesn't need any changing. The delivered spectrum.pcf looks like this:

```
** spectrum.pcf **
```

```
datafile,conABCD
```

```
data,sABCD,delete
```

```
scale,no
```

```
zerolevel,no
```

```
data-window,t-sequence,n=512
```

```
hanning,yes
```

```
series,all
```

printplot,yes
shape,no
use
poly2,calculate
end:use
par-file,sABCD
parameter
all
end:parameter

Explanation of the command lines:

A lot of command lines are the same as used in the program WAVES, so only the new command lines will be explained here.

data-window,t-sequence,n=512

The command sets the length of the subseries. Three selection types are optional, depending on the second parameter. The number of values that must be read out of a series to fill a subseries is always a power of two and the number of frequencies in the subseries than will be $2^{m-1}+1$. In t-sequence,n=x: x is the number of values that must be selected out of the series to fill a subseries. If the condition 2^m is not met the number is set at the closest higher figure that does meet this condition. The maximum number of values that can be read out of a series to fill a subseries is 2048.

hanning,yes

In the frequency domain the Hanning smoothing filter may be used. If the command is not available or NO is set the smoothing will not be used.

printplot,yes

All the computed spectral values can be presented in the output file. The values themselves are in a column and on the same line of a value a sign is printed in the plot area such that the position indicates the relation to the minimum and maximum values in the complete series. This is done for all spectral series values.

shape,no

Computed spectra are stored with there energy density values. However, it is possible to store them in a normalized way: the maximum value found is set to 1 and all other values are scaled to this maximum value. The computed parameter values are always for the non-normalized spectrum. In this way the spectra of various series can be plotted in one graph allowing a comparison of the shape of these spectra. This plotting can be done with AUKEPLOT.

4.6 Extracting results from the parameter files

Now that you're done with Delft-AUKE results can be extracted from the output of WAVES and SPECTRUM. It is also possible to calculate the averages of each column from the .txt file. All this can be done with the same excel program (*results.xls*) as the converting of the raw data.

Here you have to open the sheet '*extracting results*'. The yellow cells have to be filled with some standard values. In the first four columns the places where the parameters can be found have to be filled in. Column A has to be filled with the name of the measuring device. This has to be exactly the same as you named it with CONASC. Column B has to be filled with the name of the parameter. You

can find the parameters that have been calculated by Delft-AUKE by opening a .par file with excel. For the .txt files this is always the average. Column C is where you have to fill in a "w", "s" or "t" (WAVES, SPECTRUM or text file) for the file you want the data from. And finally in column D the column number where the parameter can be found has to be filled in. This only has to be done with the .txt files. When you have filled in all the cells the button can be clicked and the macro (Appendix B) will start.

This macro will create a new sheet within *results.xls*. In this sheet the first two rows will be the information of the parameters. The first column will be the file number. In all the other columns the data extracted from the parameter files will be stored.

5 Checklist

If you want to extract parameters from raw data, there are a couple of steps you have to take. All the steps have been mentioned above. But as an easy reminder a checklist with all the actions you have to take is given here.

1. ***Make sure you have raw data files in the comma separated value (.csv files) format.***
2. ***Check the raw data files for corrupt files and delete them.***
3. ***Check the macro security level and set it at medium.***
4. ***Open the sheet 'change file' in 'results.xls'.***
5. ***Fill in the yellow cells and click the button to run the macro.***
6. ***If you didn't take the same number of rows for every file, change the number of ending rows in 'conasc.pcf', if there are any.***
7. ***Change the get-columns command line in 'conasc.pcf' if necessary.***
8. ***Run Delft-AUKE by activating the 'start__.bat' file.***
9. ***Open the sheet 'extracting results' in results.xls.***
10. ***Fill in the yellow cells and the first columns before clicking the button to run the macro.***
11. ***You're done. The parameters will be in a new sheet in 'results.xls'***

A. Macro change files

This is the macro behind the button at sheet '*change files*'

```
Sub change_files()
```

```
,
```

```
' change_files Macro
```

```
' Macro recorded 06/03/2006 by Arjen Koopman
```

```
,
```

```
'this prevents the screen from flashing while running the macro.
```

```
Application.ScreenUpdating = False
```

```
'Activate and read the worksheet.
```

```
Worksheets("change files").Activate
```

```
dirname = Cells(1, 4)
```

```
dirnameout = Cells(2, 4)
```

```
nrow = Cells(3, 4)
```

```
ncol = Cells(4, 4) + 1
```

```
nstep = Cells(5, 4)
```

```
nhead = Cells(6, 4)
```

```
nibl = Cells(10, 4)
```

```
iii = 0
```

```
'Making a new workbook for the .bat file.
```

```
namebat = "start" & Cells(8, 4) & ".bat"
```

```
Workbooks.Add
```

```
wname = ActiveWorkbook.Name
```

```
Windows("results.xls").Activate
```

```
Worksheets("change files").Activate
```

```
'The files that should be opened are given here.
```

```
For i = 1 To Cells(9, 4)
```

```
  If i < 10 Then
```

```
    filein1 = Cells(7, 4) & "00" & i & ".csv"
```

```
    fileout1 = Cells(8, 4) & "00" & i
```

```
  Elseif i < 100 Then
```

```
    filein1 = Cells(7, 4) & "0" & i & ".csv"
```

```
    fileout1 = Cells(8, 4) & "0" & i
```

```
  Elseif i >= 100 Then
```

```
    filein1 = Cells(7, 4) & i & ".csv"
```

```
    fileout1 = Cells(8, 4) & i
```

```
  End If
```

```
  fileout = fileout1 & ".txt"
```

```
  filein = dirname & filein1
```

```
'Opening the original file.
```

```
Workbooks.OpenText Filename:=filein, Origin:= _
```

```
  xlWindows, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
```

```
  xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=True, Semicolon:=False, _
```

```
Comma:=False, Space:=False, Other:=False, FieldInfo:=Array(Array(1, 1), _  
Array(2, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1), Array(9, 1), Array(10, 1),  
Array(11, 1))
```

'Ignoring empty files.

```
If nrow = -1 Then  
    If Cells(1 + nhead, 2) = Empty Then  
        Workbooks(filein1).Close savechanges:=False  
        GoTo next_i  
    End If  
Elseif nrow <> -1 Then  
    If Cells(nrow + nhead, 2) = Empty Then  
        Workbooks(filein1).Close savechanges:=False  
        GoTo next_i  
    End If  
End If
```

change:

'Deleting the header and ending rows.

```
coll = Chr(ncol + 64)  
If nrow = -1 Then  
    Range("A" & 1 + nhead & ":" & coll & 65536).Select  
Else  
    Range("A" & 1 + nhead & ":" & coll & nrow + nhead).Select  
End If  
Selection.Copy  
Sheets.Add  
ActiveSheet.Paste  
Application.CutCopyMode = False
```

'Inserting a column in front of the data if there isn't a column with a label number.

```
If nlbl = "n" Or nlbl = "N" Then  
    Columns("A:A").Select  
    Selection.Insert Shift:=xlToRight  
End If
```

'Adding a column with the time

```
If nrow = -1 Then  
    ii = 1  
ii: If Cells(ii, 2) <> Empty Then  
    Cells(ii, 1) = (ii - 1) * nstep  
    ii = ii + 1  
    GoTo ii  
Else: GoTo save  
End If  
Else  
    For ii = 1 To nrow  
        Cells(ii, 1) = (ii - 1) * nstep  
    Next ii  
End If
```

save:

'Saving and closing the new file

```
ActiveWorkbook.SaveAs Filename:=dirnameout & fileout, _  
FileFormat:=xlCSV, CreateBackup:=False  
Workbooks(fileout).Close savechanges:=False
```

'Filling the workbook with the lines of the .bat file.

```
Windows(wname).Activate  
iii = iii + 1  
Cells(iii, 1) = "call golven.bat " & fileout1  
Windows("results.xls").Activate  
Worksheets("change files").Activate
```

'Give the filename that has just been converted in the bottom left of the screen.

```
next_i:  
Application.StatusBar = "File: " & fileout  
Next i
```

stop_it:

'Saving the .bat file.

```
Windows(wname).Activate  
ActiveWorkbook.SaveAs Filename:=dirnameout & namebat, FileFormat:=xlCSV  
Workbooks(namebat).Close savechanges:=False
```

```
Application.StatusBar = "Changing files completed"  
Application.ScreenUpdating = True
```

End Sub

B. Macro extract data

This is the macro behind the button at sheet '*extracting results*'

Sub extract_data()

'Macro recorded 20/02/2006 by Arjen Koopman

'This prevents the screen from flashing while running the macro.

Application.ScreenUpdating = False

Dim onlytext As Boolean

Dim names(100, 7)

Dim result(100)

'Adding a sheet for the results to be put in.

Worksheets("extracting results").Activate

outsheet = Cells(8, 10)

Sheets.Add

ActiveSheet.Name = outsheet

ActiveSheet.Move After:=Sheets(ActiveWorkbook.Sheets.Count)

'Activate and read the sheet 'extracting results'

Worksheets("extracting results").Activate

dirname = Cells(1, 10)

dirname2 = Cells(2, 10)

n = Cells(3, 10)

nrow = Cells(4, 10)

last = Cells(9, 10)

'Filling the names array with the information of the parameters stated in first 4

'columns of the worksheet extracting results.

For i = 1 To n

 For ii = 1 To 4

 names(i, ii) = Cells(i + 1, ii)

 Next ii

Next i

'Giving the names to the columns in the output sheet.

For ii = 1 To n

 Worksheets(outsheet).Cells(1, ii + 1) = names(ii, 1)

 Worksheets(outsheet).Cells(2, ii + 1) = names(ii, 2)

Next ii

'The files that should be opened are given here.

For i = 1 To last

 If i < 10 Then

 ccc = "00"

 Elseif i < 100 Then

 ccc = "0"

 Elseif i >= 100 Then

 ccc = Empty

```

End If
filein1 = dirname & Cells(5, 10) & ccc & i & ".par"
filein2 = dirname & Cells(6, 10) & ccc & i & ".par"
filein3 = dirname2 & Cells(7, 10) & ccc & i & ".txt"
Dsheet1 = Cells(5, 10) & ccc & i
Dsheet2 = Cells(6, 10) & ccc & i
Dsheet3 = Cells(7, 10) & ccc & i
Dbook1 = Cells(5, 10) & ccc & i & ".par"
Dbook2 = Cells(6, 10) & ccc & i & ".par"
Dbook3 = Cells(7, 10) & ccc & i & ".txt"

```

'Opening the spectrum .par file.

```

On Error Resume Next
Workbooks.OpenText Filename:=filein1, Origin:= _
xlWindows, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=False, Semicolon:=False, _
Comma:=True, Space:=True, Other:=False, FieldInfo:=Array(Array(1, 1), _
Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1))
If Err = 1004 Then GoTo skip_this

```

'Extracting data from the spectrum .par file.

```

For ii = 1 To n
  If names(ii, 3) = "s" Then
    For iii = 1 To 1000
      If Cells(iii, 2) = names(ii, 1) Then
        For iiiii = 1 To 21
          If Cells(iii + iiiii, 1) = names(ii, 2) Then
            result(ii) = Worksheets(Dsheet1).Cells(iii + iiiii, 6)
            GoTo next_ii
          End If
        Next iiiii
      End If
    Next iii
  End If
Next ii
next_ii:
Next ii

```

'Opening the waves .par file.

```

Workbooks.OpenText Filename:=filein2, Origin:= _
xlWindows, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=False, Semicolon:=False, _
Comma:=True, Space:=True, Other:=False, FieldInfo:=Array(Array(1, 1), _
Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1))

```

'Extracting data from the waves .par file.

```

For ii = 1 To n
  If names(ii, 3) = "w" Then
    For iii = 1 To 1000
      If Cells(iii, 2) = names(ii, 1) Then
        For iiiii = 1 To 15
          If Cells(iii + iiiii, 1) = names(ii, 2) Then
            result(ii) = Worksheets(Dsheet2).Cells(iii + iiiii, 6)
          End If
        Next iiiii
      End If
    Next iii
  End If
Next ii

```

```

        GoTo nextii
    End If
Next iii
End If
Next iii
End If
nextii:
Next ii

```

'Opening the .txt file.

```

Workbooks.OpenText Filename:=filein3, Origin:= _
xlWindows, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=False, Semicolon:=False, _
Comma:=True, Space:=True, Other:=False, FieldInfo:=Array(Array(1, 1), _
Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1))

```

'Extracting data from the .txt file

```

For ii = 1 To n
    If names(ii, 3) = "t" Then
        Sum = 0
        coltxt = names(ii, 4)
        For iii = 1 To nrow
            Sum = Sum + Worksheets(Dsheet3).Cells(iii, coltxt)
        Next iii
        Average = Sum / nrow
        result(ii) = Average
    End If
Next ii

```

'Closing the .par and .txt files

```

Workbooks(Dbook1).Close savechanges:=False
Workbooks(Dbook2).Close savechanges:=False
Workbooks(Dbook3).Close savechanges:=False

```

'Adding the number of the file to the first column of the output sheet.

```

Worksheets(outsheet).Cells(i + 2, 1) = i

```

'Filling in all the results in the output sheet.

```

For ii = 1 To n
    Worksheets(outsheet).Cells(i + 2, ii + 1) = result(ii)
Next ii
Erase result()

```

```

GoTo next_i

```

'If an error occurred the error number is given.

```

skip_this:
    Worksheets(outsheet).Cells(i + 2, 1) = i
    Worksheets(outsheet).Cells(i + 2, 2) = "error: " & Err.Number

```

'With this line you can see in the statusbar what the macro is doing, while it's running.

```

next_i:

```

```
Application.StatusBar = "File: " & Dsheet1  
Next i
```

```
stop_it:  
Application.StatusBar = "Extraction completed"  
Application.ScreenUpdating = True
```

```
End Sub
```